STIC Search Report
EIC 2100

STIC Database Tracking Number: 111635

TO: Thomas J Cleary **Location: 2A49** **Art Unit : 2111** **Thursday, January 08, 2004** **Case Serial Number: 09/870447**	**From: David Holloway** **Location: EIC 2100** **PK2-4B30** **Phone: 308-7794** **david.holloway@uspto.gov**

Search Notes

Dear Examiner Cleary,

Attached please find your search results for above-referenced case.
Please contact me if you have any questions or would like a re-focused search.

David

```
Set      Items   Descript
S1      258917   INTERRUPT? OR EXCEPTION? OR ISR OR TRAP? ?
S2      997718   SCHEDUL? OR TIME OR TIMING OR TIMED OR INTERVAL OR DURATION
              OR (NUMBER OR COUNT? OR QUANTIT?)(N)(CYCLE?)
S3     1637689   MODIF? OR SET OR SETS OR CHANG? OR VARY OR VARIABLE?
S4      788264   DISABL? OR INHIBIT? ? OR STOP OR PREVENT? OR STOPPING
S5        4300   S1 (10N) S2 (10N) S3 AND S4
S6      677276   USER? OR INDIVIDUAL? OR PERSONAL? OR OPERATOR? OR ADMINIST-
             RATOR?
S7        1663   S5(S)S6
S8         682   S7(S)(CPU OR CPUS OR MICROPROCESSOR? OR PROCESSOR? OR COMP-
             UTER? OR WORKSTATION? OR PLC OR PROGRAMMABLE()LOGIC()CONTROLL-
             ER? OR WORK()STATION?)
S9         343   S8 AND IC=G06F?
S10       7075   S1(4N)S4
S11      88583   S3(4N)S6
S12        105   S9 AND S10 AND S11
S13         40   S12 AND IC=G06F-009?
File 348:EUROPEAN PATENTS 1978-2003/Dec W02
         (c) 2003 European Patent Office
File 349:PCT FULLTEXT 1979-2002/UB=20031225,UT=20031218
         (c) 2003 WIPO/Univentio
```

13/5,K/13 (Item 13 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00442636
Prolog interrupt processing.
Prolog-Unterbrechungsverarbeitungssystem.
Systeme de traitement d'interruptions en Prolog.
PATENT ASSIGNEE:
International Business Machines Corporation, (200120), Old Orchard Road, Armonk, N.Y. 10504, (US), (applicant designated states: DE;FR;GB)
INVENTOR:
Gillet, Marc Jean Louis, 1272 Woodside Road, Redwood City, CA 94061, (US)
LEGAL REPRESENTATIVE:
Burt, Roger James, Dr. et al (52152), IBM United Kingdom Limited Intellectual Property Department Hursley Park, Winchester Hampshire SO21 2JN, (GB)
PATENT (CC, No, Kind, Date): EP 414651 A1 910227 (Basic)
APPLICATION (CC, No, Date): EP 90850251 900626;
PRIORITY (CC, No, Date): US 393722 890814
DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS: **G06F-009/44 ; G06F-009/46**
CITED REFERENCES (EP A):
IBM TECHNICAL DISCLOSURE BULLETIN vol. 27, no. 10B, March 1985, pages 6036,6037, Armonk, NY, US; "Soft Stopping for System Processors"
IBM TECHNICAL DISCLOSURE BULLETIN vol. 27, no. 10B, March 1985, page 6045, Armonk, NY, US; "Picocode Interrupt Mechanism";

ABSTRACT EP 414651 A1
An improved architecture for a Prolog interpreter/compiler is described to facilitate interrupt processing. The new architecture employs interrupt control words and a set of interrupt control blocks to enhance Prolog processing. As interrupts are detected, an interrupt control word is loaded with the address of an interrupt control block for managing interrupt processing. Then, each time a new predicate is fired, the interrupt control word is tested and control is passed to the interrupt processing routine if the control word is non-zero.
ABSTRACT WORD COUNT: 86

LEGAL STATUS (Type, Pub Date, Kind, Text):
Application: 910227 A1 Published application (A1with Search Report ;A2without Search Report)
Examination: 910227 A1 Date of filing of request for examination: 901213
Change: 930324 A1 Representative (change)
Examination: 931215 A1 Date of despatch of first examination report: 931103
Withdrawal: 940928 A1 Date on which the European patent application was withdrawn: 940801
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPABF1	362
SPEC A	(English)	EPABF1	13964
Total word count - document A			14326
Total word count - document B			0
Total word count - documents A + B			14326

INTERNATIONAL PATENT CLASS: **G06F-009/44** ...

... **G06F-009/46**

...SPECIFICATION or customized by a user Prolog program. The best analogy to a Daemon in classical **computer** processing is a task. A task is a set of instructions often for a single...

...may feed information to a suspended task. A Prolog Daemon is initiated

to process the **interr** during the designated **time** tween predicates. Some examples of logical **interrupt** processing are provided below.

Delayed Evaluation

Predefined PROLOG logical **interrupts** include delayed evaluation. Delayed evaluation is the instanciation of a frozen **variable** . A frozen **variable** is a **variable** awaiting the completion of another goal to complete its processing. The Daemon is the waiting...

...A frozen variable is a special unbound variable. When a value is given to this **variable** , a **user** defined Prolog predicate is triggered.

GARBAGE COLLECTION

The logical interrupt is the global of trail...interrupts sensitive to backtrack and unsensitive to backtrack. For example, if there is a Prolog **interrupt** associated with system command **Stop** Predicate (SP), and if the predicate Prolog associated to the **interrupt** is defined as ' **stop** the execution of the current command', then after someone has entered the SP command, the...language independent because the variations for individual languages are placed in the data structures.

No **modification** to **user** programs is necessary when a new release of a language becomes available. Changes are only...

```
Set     Items   Descript
S1    1321505   INTERRUPT? OR EXCEPTION? OR ISR OR TRAP? ?
S2   12685235   SCHEDUL? OR TIME OR TIMING OR TIMED OR INTERVAL OR DURATION
              OR (NUMBER OR COUNT? OR QUANTIT?)(N)(CYCLE?)
S3       8333   S1(10N)S2(10N)(MODIF? OR SET OR SETS OR CHANG? OR VARY OR -
              VARIAB?)
S4       9828   S1(5N)(PREVENT? OR STOP? ? OR INHIBIT? OR DISABL? OR STOPP-
              ING OR HALT??? ?)
S5        121   S3(S)S4
S6         86   RD (unique items)
S7         64   S6 AND (USER? OR INDIVIDUAL? OR OPERATOR? OR PERSONAL? OR -
              ADMINISTRATOR?)
S8         61   S7 NOT PY>2001
S9         58   S8 NOT PD=20010502:20040111
S10        31   S9 AND (CPU OR CPUS OR PLC OR PLCS OR PROGRAMMABLE()LOGIC(-
              )CONTROLLER? OR WORKSTATION? OR PROCESSOR? OR MICROPROCESSOR?
              OR WORK()STATION?)
File 148:Gale Group Trade & Industry DB 1976-2004/Jan 08
         (c)2004 The Gale Group
File 647:CMP  Computer Fulltext 1988-2004/Dec W4
         (c) 2004 CMP Media, LLC
File  98:General Sci Abs/Full-Text 1984-2003/Nov
         (c) 2003 The HW Wilson Co.
File   9:Business & Industry(R) Jul/1994-2004/Jan 07
         (c) 2004 Resp. DB Svcs.
File  15:ABI/Inform(R) 1971-2004/Jan 07
         (c) 2004 ProQuest Info&Learning
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 369:New Scientist 1994-2003/Dec W2
         (c) 2003 Reed Business Information Ltd.
File 674:Computer News Fulltext 1989-2003/Dec W3
         (c) 2003 IDG Communications
File 621:Gale Group New Prod.Annou.(R) 1985-2004/Jan 08
         (c) 2004 The Gale Group
File 484:Periodical Abs Plustext 1986-2004/Jan W1
         (c) 2004 ProQuest
File 624:McGraw-Hill Publications 1985-2004/Jan 07
         (c) 2004 McGraw-Hill Co. Inc
File  16:Gale Group PROMT(R) 1990-2004/Jan 08
         (c) 2004 The Gale Group
File 636:Gale Group Newsletter DB(TM) 1987-2004/Jan 08
         (c) 2004 The Gale Group
File 275:Gale Group Computer DB(TM) 1983-2004/Jan 08
         (c) 2004 The Gale Group
File  47:Gale Group Magazine DB(TM) 1959-2004/Dec 31
         (c) 2004 The Gale group
```

```
Set     Items   Descript
S1     448484   INTERRUPT? OR EXCEPTION? OR ISR OR TRAP? ?
S2    5940973   SCHEDUL? OR TIME OR TIMING OR TIMED OR INTERVAL OR DURATION
             OR (NUMBER OR COUNT? OR QUANTIT?)(N)(CYCLE?)
S3    8344492   MODIF? OR SET OR SETS OR CHANG? OR VARY OR VARIABLE?
S4    2936622   USER? OR OPERATOR? OR INDIVIDUAL? OR PERSONAL? OR ADMINIST-
            RATOR?
S5    2748633   DISABL? OR INHIBIT? OR PREVENT? OR STOP OR STOPPING
S6       2082   S1 AND S2 AND S3 AND S5
S7      76677   S4(3N)S3
S8       2226   S1(3N)S5
S9          1   S2 AND S7 AND S8
S10       295   S4 AND S6
S11        39   S10 AND (S7 OR S8)
S12        33   RD (unique items)
S13        29   S12 NOT PY>2001
S14        14   S10 AND (MICROPROCESSOR? OR PROCESSOR? OR CPU OR CPUS OR P-
              LC OR PROGRAMMABLE()LOGIC()CONTROLLER? OR PC OR PCS OR WORKST-
              ATION? OR WORK()STATION?)
S15        10   S14 NOT S11
S16         6   RD (unique items)
File   8:Ei Compendex(R) 1970-2004/Dec W4
         (c) 2004 Elsevier Eng.  Info. Inc.
File  35:Dissertation Abs Online 1861-2003/Nov
         (c) 2003 ProQuest Info&Learning
File 202:Info. Sci. & Tech. Abs. 1966-2003/Nov 17
         (c) 2003 EBSCO Publishing
File  65:Inside Conferences 1993-2004/Jan W1
         (c) 2004 BLDSC all rts. reserv.
File   2:INSPEC 1969-2003/Dec W2
         (c) 2003 Institution of Electrical Engineers
File  94:JICST-EPlus 1985-2004/Dec W4
         (c)2004 Japan Science and Tech Corp(JST)
File 111:TGG Natl.Newspaper Index(SM) 1979-2004/Jan 06
         (c) 2004 The Gale Group
File 233:Internet & Personal Comp. Abs. 1981-2003/Sep
         (c) 2003 EBSCO Pub.
File   6:NTIS 1964-2004/Jan W1
         (c) 2004 NTIS, Intl Cpyrght All Rights Res
File 144:Pascal 1973-2003/Dec W2
         (c) 2003 INIST/CNRS
File  34:SciSearch(R) Cited Ref Sci 1990-2003/Dec W4
         (c) 2003 Inst for Sci Info
File  99:Wilson Appl. Sci & Tech Abs 1983-2003/Nov
         (c) 2003 The HW Wilson Co.
```

[redacted]3/5/17 (Item 4 from [redacted]ile: 2)
[redacted]IALOG(R)File 2:INSPEC
[redacted](c) 2003 Institution of Electrical Engineers. All rts. reserv.

4758570 INSPEC Abstract Number: C9410-6150G-018
Title: Software abort and multiprocessor debugging
Author(s): Baek Youngsik; Jin Sungil
Author Affiliation: Dept. of Comput. Sci., Chung Nam Nat. Univ., Daejeon, South Korea
Part vol.1 p.237-41 vol.1
Editor(s): Yuan Baozong
Publisher: IEEE, New York, NY, USA
Publication Date: 1993 Country of Publication: USA 5 vol. (xxvi+1206+xvii+676+xv+580+xvii+619) pp.
ISBN: 0 7803 1233 3
Conference Title: Proceedings of TENCON '93. IEEE Region 10 International Conference on Computers, Communications and Automation
Conference Date: 19-21 Oct. 1993 Conference Location: Beijing, China
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: Halting all processes distributed machines, simultaneously, is difficult due to the communication delay. This problem can be solved or alleviated by existing halting algorithms, which however cannot avoid a minimal **change** when executing **timing** sensitive programs. It is important, especially in detecting **time** -dependent errors, that the debugger be able to **stop** a **set** of processes as fast as possible in order to preserve the relative states of the programmer inspection. This paper suggests a methodology and its hardware implementation, Conductor, as a laboratory model. This methodology enables the **user** to use a local debugging tool without **modifications** . So, the **user** can debug distributed processes in a multiprocessor debugging environment in a similar way to a single processor environment. By using Conductor, the **user** can **set** up breakpoints in distributed processes, and **stop** almost simultaneously at an instruction at an appropriate point where the **user** wants to see the status of the processes. It preserves relative consistency among processes and is important in inspecting **time** -dependent errors. (8 Refs)
Subfile: C
Descriptors: interactive systems; **interrupts** ; multiprocessing programs; program debugging
Identifiers: multiprocessor debugging; software abort; distributed machines; communication delay; halting algorithms; **timing** sensitive programs; relative states; Conductor; local debugging tool; breakpoints; relative consistency
Class Codes: C6150G (Diagnostic, testing, debugging and evaluating systems); C6150N (Distributed systems)

1 6/5/6 (Item 3 from [redacted]le: 2)
DIALOG(R)File 2:INSPEC

5268037 INSPEC Abstract Number: C9606-6150N-098
Title: Eliminating receive livelock in an interrupt **-driven kernel**
Author(s): Mogul, J.C.; Ramakrishnan, K.
Author Affiliation: Western Res. Lab., Digital Equipment Corp., Palo Alto, CA, USA
Conference Title: Proceedings of the USENIX 1996 Annual Technical Conference p.99-111
Publisher: USENIX Assoc, Berkeley, CA, USA
Publication Date: 1996 Country of Publication: USA 352 pp.
Material Identity Number: XX96-00120
Conference Title: Proceedings of USENIX
Conference Date: 22-26 Jan. 1996 Conference Location: San Diego, CA, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: Most operating systems use interface **interrupts** to **schedule** network tasks. **Interrupt** -driven systems can provide low overhead and good latency at low offered load, but degrade significantly at higher arrival rates unless care is taken to **prevent** several pathologies. These are various forms of receive livelock, in which the system spends all its **time** processing **interrupts** , to the exclusion of other necessary tasks. Under extreme conditions, no packets are delivered to the **user** application or the output of the system. To avoid livelock and related problems, an operating system must **schedule** network **interrupt** handling as carefully as it **schedules** process execution. We **modified** an **interrupt** -driven networking implementation to do so; this eliminates receive livelock without degrading other aspects of system performance. We present measurements demonstrating the success of our approach. (16 Refs)
Subfile: C
Descriptors: **interrupts** ; operating system kernels; **processor scheduling** ; Unix
Identifiers: receive livelock elimination; **interrupt** -driven kernel; operating systems; interface **interrupts** ; network task **scheduling** ; overhead; latency; arrival rates; **interrupt** processing; **user** application; network **interrupt** handling **scheduling** ; **interrupt** -driven networking implementation; system performance
Class Codes: C6150N (Distributed systems software); C6150J (Operating systems)

```
Set     Items   Description
S1     233436   INTERRUPT? OR EXCEPTION? OR ISR OR TRAP? ?
S2    2870551   SCHEDUL? OR TIME OR TIMING OR TIMED OR INTERVAL OR DURATION
             OR (NUMBER OR COUNT? OR QUANTIT)(N)(CYCLE?)
S3    3416684   MODIF? OR SET OR SETS OR CHANG? OR VARY OR VARIABLE?
S4    3833856   DISABL? OR INHIBIT? ? OR BLOCK? OR STOP OR PREVENT? OR STO-
            PPING
S5       7502   S1 AND S2 AND S3 AND S4
S6      12968   S1(5N)S4
S7       1344   S5 AND S6
S8        355   S7 AND IC=G06F?
S9        260   S8 AND (MICROPROCESS? OR PROCESSOR? OR PC OR COMPUTER? OR -
               WORKSTATION? OR WORK()STATION? OR CPU OR CPUS)
S10       253   S9 NOT AD>20010502
S11      1034   S6(10N)S3(10N)S4
S12        85   S10 AND S11
S13        72   S12 NOT BLOCK? ?
S14        27   S10 AND (USER? ? OR INDIVIDUAL? OR PERSONAL? OR OPERATOR?)
S15        18   S14 NOT S13
File 347:JAPIO Oct 1976-2003/Sep(Updated 040105)
         (c) 2004 JPO & JAPIO
File 350:Derwent WPIX 1963-2004/UD,UM &UP=200402
         (c) 2004  Thomson Derwent
```

13/5/59 (Item 10 from file: 350)
DIALOG(R)File 350:Derwent WPIX

010837168 **Image available**
WPI Acc No: 1996-334121/199633
XRPX Acc No: N96-281531

Dynamic user interrupt **scheme in programmable logic controller, e.g for petrol pumps - allows for de-assignment of user program sections in order to allow user program to transfer control upon occurrence of event of interest**

Patent Assignee: SIEMENS IND AUTOMATION INC (SIEI); SIEMENS ENERGY & AUTOMATION INC (SIEI); SIEMENS ENERGY & AUTOMATION (SIEI)
Inventor: BOGGS M; FULTON T; MITCHELL R; PALERMO R J
Number of Countries: 020 Number of Patents: 007
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 9621180	A1	19960711	WO 95US8727	A	19950712	199633	B
EP 800670	A1	19971015	EP 95926256	A	19950712	199746	
			WO 95US8727	A	19950712		
US 5765000	A	19980609	US 94365658	A	19941229	199830	
EP 800670	B1	19990421	EP 95926256	A	19950712	199920	
			WO 95US8727	A	19950712		
DE 69509262	E	19990527	DE 609262	A	19950712	199927	
			EP 95926256	A	19950712		
			WO 95US8727	A	19950712		
ES 2135075	T3	19991016	EP 95926256	A	19950712	199950	
CN 1171850	A	19980128	CN 95197171	A	19950712	200328	
			WO 95US8727	A	19950712		

Priority Applications (No Type Date): US 94365658 A 19941229
Cited Patents: No-Citns.
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 9621180	A1	E	46	G05B-019/05	
Designated States (National): BR CA CN					
Designated States (Regional): AT BE CH DE DK ES FR GB GR IE IT LU MC NL PT SE					
EP 800670	A1	E		G05B-019/05	Based on patent WO 9621180
Designated States (Regional): DE ES FR GB IT					
US 5765000	A			G06F-013/26	
EP 800670	B1	E		G05B-019/05	Based on patent WO 9621180
Designated States (Regional): DE ES FR GB IT					
DE 69509262	E			G05B-019/05	Based on patent EP 800670
					Based on patent WO 9621180
ES 2135075	T3			G05B-019/05	Based on patent EP 800670
CN 1171850	A			G05B-019/05	Based on patent WO 9621180

Abstract (Basic): WO 9621180 A

The dynamic user **interrupt** scheme comprises a programmable logic controller (PLC) having at least one input and one output terminal for transmitting and receiving respectively predetermined signals. At least one **microprocessor** is contained in the PIC for executing user specified primary calculations and commands which are performed according to a predetermined cyclical order.

Interrupts cooperate with the **microprocessor** for **interrupting** the predetermined cyclical order and **stopping** the **microprocessor** from executing the user specified primary calculations and commands and for performing at least one secondary **set** of calculations and commands. The **interrupts** initiate the **interruption** in response to a predetermined user specified condition. **Interrupt** routines may be dynamically reassigned during program execution rather than at compilation **time** and may be automatically performed in order to effectuate control transfer.

USE/ADVANTAGE - For printers, beeper paging. Allows for dynamically presetting values of characters and the like as well as pipelining of **interrupts** in the PLC.

Dwg.9/12

Title Terms: DYNAMIC; USER; **INTERRUPT** ; SCHEME; PROGRAM; LOGIC; CONTROL; GASOLINE; PUMP; ALLOW; DE; ASSIGN; USER; PROGRAM; SECTION; ORDER; ALLOW; USER; PROGRAM; TRANSFER; CONTROL; OCCUR; EVENT; INTEREST
Derwent Class: T01; T06; X25
International Patent Class (Main): G05B-019/05; **G06F-013/26**
International Patent Class (Additional): **G06F-009/46**
File Segment: EPI

15/5/8 (Item 8 from file: 347)
DIALOG(R)File 347:JAPIO

01765249 **Image available**
METHOD FOR CHECKING OPERATION OF WATCHDOG TIMER

PUB. NO.: 60-243749 [JP 60243749 A]
PUBLISHED: December 03, 1985 (19851203)
INVENTOR(s): KISA KAZUYUKI
APPLICANT(s): FUJI ELECTRIC CO LTD [000523] (A Japanese Company or Corporation), JP (Japan)
APPL. NO.: 59-097556 [JP 8497556]
FILED: May 17, 1984 (19840517)
INTL CLASS: [4] **G06F-011/30**
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units)
JAPIO KEYWORD:R131 (INFORMATION PROCESSING -- Microcomputers & **Microprocessers**)
JOURNAL: Section: P, Section No. 452, Vol. 10, No. 116, Pg. 1, April 30, 1986 (19860430)

ABSTRACT

PURPOSE: To detect the fault of a watchdog timer itself and to detect surely abnormality including program runaway by checking the operation of the watchdog timer when a control device is in a test mode.

CONSTITUTION: When the watchdog timer operates normally, a mask **disabling interruption** (**TRAP**) is generated after a prescribed period and a program B is started. In a testing **time** , the program B informs the input of the **interruption** to a program A. When a counter is not overflowed and the **TRAP** is received from a program B, the program A compares the counted value obtained up to the point of **time** with a prescribed **set** point, and if the compared result is included in a prescribed range, decides that the watchdog timer is normal. If the counter is overflowed or the **time** up to the reception of the **TRAP** exceeds the prescribed range, the watchdog timer is regarded as abnormal status and the result is informed to the **operator** or the like.

15/5/10 (Item 2 from file: 350)
DIALOG(R)File 350:Derwent WPIX

013895893 **Image available**
WPI Acc No: 2001-380106/200140
XRPX Acc No: N01-278599
Code reading method for nonvolatile write able memory, involves suspending non-read operation and enabling interrupts **for reading code from memory**
Patent Assignee: INTEL CORP (ITLC)
Inventor: HASBUN R N; SEE D L
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6189070	B1	20010213	US 97919539	A	19970828	200140	B

Priority Applications (No Type Date): US 97919539 A 19970828
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6189070	B1		15	G06F-013/00	

Abstract (Basic): US 6189070 B1
NOVELTY - **Interrupts** are **disabled** and non-read operation such as writing is initiated by low level routines downloaded to volatile memory. In response to detecting an **interrupt** , non-read operation is suspended and memory is **set** to read mode, **interrupt** is enabled using task switches and code is read from memory.
DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:
(a) System with nonvolatile write able memory;
(b) **Computer** readable medium storing instructions for reading codes from nonvolatile memory
USE - For use in nonvolatile write able memories such as flash memory, electrically erasable programmable read only memory (EEPROMS), ferroelectric random access memory (FRAM) used in **personal computer** system, cellular phone.
ADVANTAGE - Memory status may be provided automatically during read operation and **interrupt** latency is reduced by controlling **interrupt** operation to be performed with predetermined **time** .
DESCRIPTION OF DRAWING(S) - The figure shows **block** diagram of command register and memory array control circuitry.
pp; 15 DwgNo 6/7
Title Terms: CODE; READ; METHOD; WRITING; ABLE; MEMORY; SUSPENSION; NON; READ; OPERATE; ENABLE; **INTERRUPT** ; READ; CODE; MEMORY
Derwent Class: T01
International Patent Class (Main): **G06F-013/00**
File Segment: EPI

15/5/11 (Item 3 from file: 350)
DIALOG(R)File 350:Derwent WPIX

012672170 **Image available**
WPI Acc No: 1999-478277/199940
XRPX Acc No: N99-356011
User **programmable** interrupt **mask with** time **out implementing system for** interrupt **processing in** computer
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: TEMPLE J L
Number of Countries: 003 Number of Patents: 003
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5937199	A	19990810	US 97868352	A	19970603	199940	B
KR 99006454	A	19990125	KR 9817557	A	19980515	200014	
TW 364100	A	19990711	TW 98102572	A	19980223	200030	

Priority Applications (No Type Date): US 97868352 A 19970603
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5937199	A		15	G06F-012/14	
KR 99006454	A			G06F-009/00	
TW 364100	A			G06F-009/46	

Abstract (Basic): US 5937199 A

NOVELTY - An **interrupt** controller (200) coupled to a RISC processing unit, signals an error, enables **interrupt** requests and halts the counting, when the **duration** of **disabling** the **interrupt** requests exceeds a predetermined **interval** prior to the halting of the counting.

DETAILED DESCRIPTION - A master mask (206) **disables** the **interrupt** requests to the RISC processing unit. A **time** out counter coupled to the master mask, counts the **duration** of **disabling** of **interrupt** requests by the master mask. A clocking unit is coupled to the **time** out counter for stepping the count. A comparator logic unit is also coupled to the **time** out counter for comparing the count to the **set time** value to determine if the **time** out counter has counted past **set time** period.

USE - For implementing **user** programmable **interrupt** mask with **time** out for enhancing resource locking used in field of **interrupt** processing in **computer** system.

ADVANTAGE - Re-enables **interrupt** requests if the predetermined **time** period has expired without having the **interrupt** requests re-enabled by the **user** thereby permitting the **user** to mask **interrupts** for a controlled period of **time** by **stopping** counting, when **interrupt** requests **disabling duration** exceeds predetermined **interval** .

DESCRIPTION OF DRAWING(S) - The figure shows an **interrupt** control unit.

Interrupt controller (200)
Master mask (206)
pp; 15 DwgNo 2/6

Title Terms: **USER** ; PROGRAM; **INTERRUPT** ; MASK; **TIME** ; IMPLEMENT; SYSTEM; **INTERRUPT** ; PROCESS; **COMPUTER**
Derwent Class: T01
International Patent Class (Main): **G06F-009/00** ; **G06F-009/46** ; **G06F-012/14**
File Segment: EPI